

Interface Design for Interoperability for the Land
Information System
Submitted under Task Agreement GSFC-CT-2
Cooperative Agreement Notice (CAN)
CAN-00OES-01
Increasing Interoperability and Performance of
Grand Challenge
Applications in the Earth, Space, Life, and
Microgravity Sciences

History:

Revision	Summary of Changes	Date
1.0	Draft	01/06/03
2.0	Revised with CT team comments	03/20/03

Contents

1	Introduction	4
2	Land Surface Modeling in LIS	4
3	Internal Interoperability in LIS	6
4	External Code Interoperability in LIS	8
4.1	ESMF	8
4.2	ALMA Interfaces in LIS	10
4.2.1	Input Interface	11
4.2.2	ALMA Wrappers	13
4.2.3	Output Interface	13
5	Final Remarks	14

List of Figures

1	Structure of LDAS	5
2	Structure of modules in LDAS driver	7
3	Example of introducing a new LSM(VIC) in LIS	8
4	Example of introducing a new forcing(ECMWF) in LIS	9
5	Modules <code>baseforcing_module</code> and <code>lsm_module</code>	10
6	A skeletal example of applications in ESMF	11
7	Proposed ALMA interfaces in LIS	12

List of Tables

1	Mapping of Forcing variables to NOAH input variables	13
2	Mapping of ALMA and VIC output variables : General Energy Balance	14
3	Mapping of ALMA and VIC output variables : General Water Balance	15
4	Mapping of ALMA and VIC output variables : Surface State Variables	15
5	Mapping of ALMA and VIC output variables : SubSurface State Variables	16
6	Mapping of ALMA and VIC output variables : Evaporation Variables	16
7	Mapping of ALMA and VIC output variables : Other hydrologic Variables	16
8	Mapping of ALMA and VIC output variables : Cold Season Processes	17

9	Mapping of ALMA and VIC output variables : Variables to be compared with remote sensed data	17
---	---	----

1 Introduction

This document describes the design policy for interoperability for the Land Information System (LIS) [6] implemented under funding from NASA’s ESTO Computational Technologies Project. This design is submitted to satisfy the Task Agreement GSFC-CT-2 under Cooperative Agreement Notice CAN-00-OES-01 increasing interoperability and performance of grand challenge applications in the earth, space, life, and microgravity sciences.

Code interoperability is important not only between components of a research application, but also between different applications, to decrease the cost of development. Research applications with reusable components facilitate faster development of future applications and enables a broader user base.

This document outlines two different types of interoperability that LIS intends to define and adopt:

- **Internal Interoperability:** This is interoperability that is provided by LIS to the land surface modeling community. LIS will provide an interoperable framework for the land surface modeling community by defining adaptive, extensible interfaces for incorporating new land surface models into LIS.
- **External Interoperability:** Participate with Earth, space, life, and microgravity scientific communities by adopting the utilities and compliance guidelines provided by the earth system modeling framework (ESMF) [4]. LIS will also comply with established land surface modeling standards such as assistance for land modeling activities (ALMA) [1].

2 Land Surface Modeling in LIS

In general, land surface modeling seeks to predict the terrestrial water, energy, and biogeochemical processes by solving the governing equations of soil-vegetation-snowpack medium. Land surface modeling combined with data assimilation seeks to synthesize data and land surface models to improve our ability to predict and understand these processes. The ability to predict terrestrial water, energy, and biogeochemical processes is critical for applications in weather and climate prediction, agricultural forecasting, water resources management, hazard mitigation and mobility assessment.

In order to predict water, energy, and biogeochemical processes using (typically 1-D vertical) partial differential equations, land surface models require three types of inputs: (1) initial conditions, which describe the initial state of land surface; (2) boundary conditions, which describe both the upper (atmospheric) fluxes or states, also known as “forcings” and also the lower(soil) fluxes or states; and (3) parameters,

which are a function of soil, vegetation, topography, etc., and are used to solve the governing equations.

LIS uses the LDAS [5] model control and input/output system that drives multiple offline one dimensional land surface models (LSMs) to facilitate global land surface modeling within a data assimilation system framework. LIS is expected to include three different land surface models, namely, CLM [3], NOAH [7], and VIC [8]. The LDAS driver in LIS uses various satellite and ground based observation systems within a land data assimilation framework to produce optimal output fields of land surface states and fluxes. In addition to being forced with real time output from numerical prediction models and satellite radar precipitation measurements, LDAS derives model parameters from existing topography, vegetation and soil coverages. The model results are aggregated to various temporal and spatial scales, e.g., 3 hourly, $0.25^\circ \times 0.25^\circ$.

The execution of LDAS starts with reading in the user specifications. The user selects the model domain and spatial resolution, the duration and timestep of the run, the land surface model, the type of forcing from a list of model and observation-based data sources, the number of “tiles” per grid square, the soil parameterization scheme, reading and writing of restart files, output specifications, and the functioning of several other enhancements including elevation correction and data assimilation. The LSMs in LDAS are driven by atmospheric forcing data such as precipitation, radiation, wind speed, humidity, etc., from various sources. LDAS applies spatial interpolation to convert forcing data to the appropriate resolution required by the model. Since the forcing data is read in at regular intervals, LDAS also temporally interpolates time average or instantaneous data to that needed by the model at the current time step. Figure1 shows the structure of LDAS.

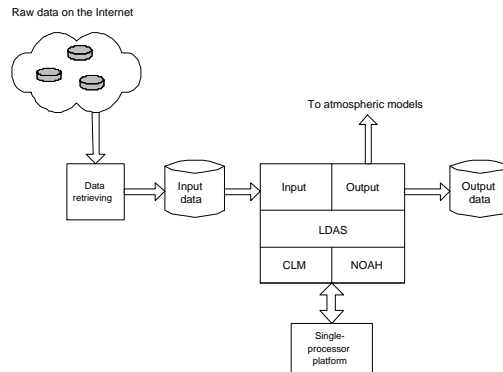


Figure 1: Structure of LDAS

3 Internal Interoperability in LIS

The concept of "internal" interoperability is to provide an interoperable framework for the land surface modeling community by defining adaptive, extensible interfaces for incorporating new land surface models into LIS. This is interoperability that is provided by LIS to the land surface modeling community to facilitate adoption of new or improved land surface models and input data.

This is achieved by reorganizing the central driver of LIS, the Land Data Assimilation System (LDAS). The LDAS driver is redesigned making use of the advanced features of Fortran 90 programming language, which are especially useful for object oriented programming. The modified driver is designed using object oriented design principles, providing a number of well-defined interfaces or "hook points" for enabling rapid prototyping and development of new features and applications into LIS.

Figure 2 shows the organization of modules and the main driver in LDAS. The modules shown in Figure 2 are designed to provide abstractions for the main functionalities in LIS. For example, `lsm_module` is designed to capture the main functionalities associated with the operation of a land surface model. This module contains interfaces and subroutines required for initialization, execution, restart and managing output of an LSM.

The main driver that initializes other modules is represented by `ldasdrv`. `ldasdrv` initializes parallelization routines through `pool_module` and the land surface modeling routines through `ldasdrv_module`. `ldas_module` contains the variables for LSM initializations, executions and outputs. The representation and management of time is encompassed in `time_module` and `grid_module` contains the variables used for spatial grid representation. `baseforcing_module` includes interfaces that are used to incorporate different atmospheric and observation forcings. As explained earlier, `lsm_module` provides interfaces that can be extended to incorporate new LSMs.

In order to define well-defined interfaces that facilitates extensibility for additional features, it is necessary to delegate the flow of control to a number of explicit interfaces and routines. Figure 4 shows an example of using the "hook points" for incorporating a new forcing scheme. The relevant methods associated with a new forcing scheme incorporation is encapsulated in the `baseforcing_module`. The call to initialize base forcing is delegated to the `baseforcing_module` from `ldasdrv` and `ldasdrv_module`. Similarly Figure 3 shows an example of incorporating a new LSM. A call to execute an LSM run is transferred to `lsm_module` from `ldasdrv` and `ldasdrv_module`. Figure 5 shows the interfaces and routines defined in `baseforcing_module` and `lsm_module`. The `get_baseforcing` routine in `baseforcing_module` is used to call the appropriate forcing such as `getgeos` or `getgdas` etc., at run time. In a similar fashion, the `run_lsm` method delegates the LSM execution to run models such as CLM, NOAH or VIC.

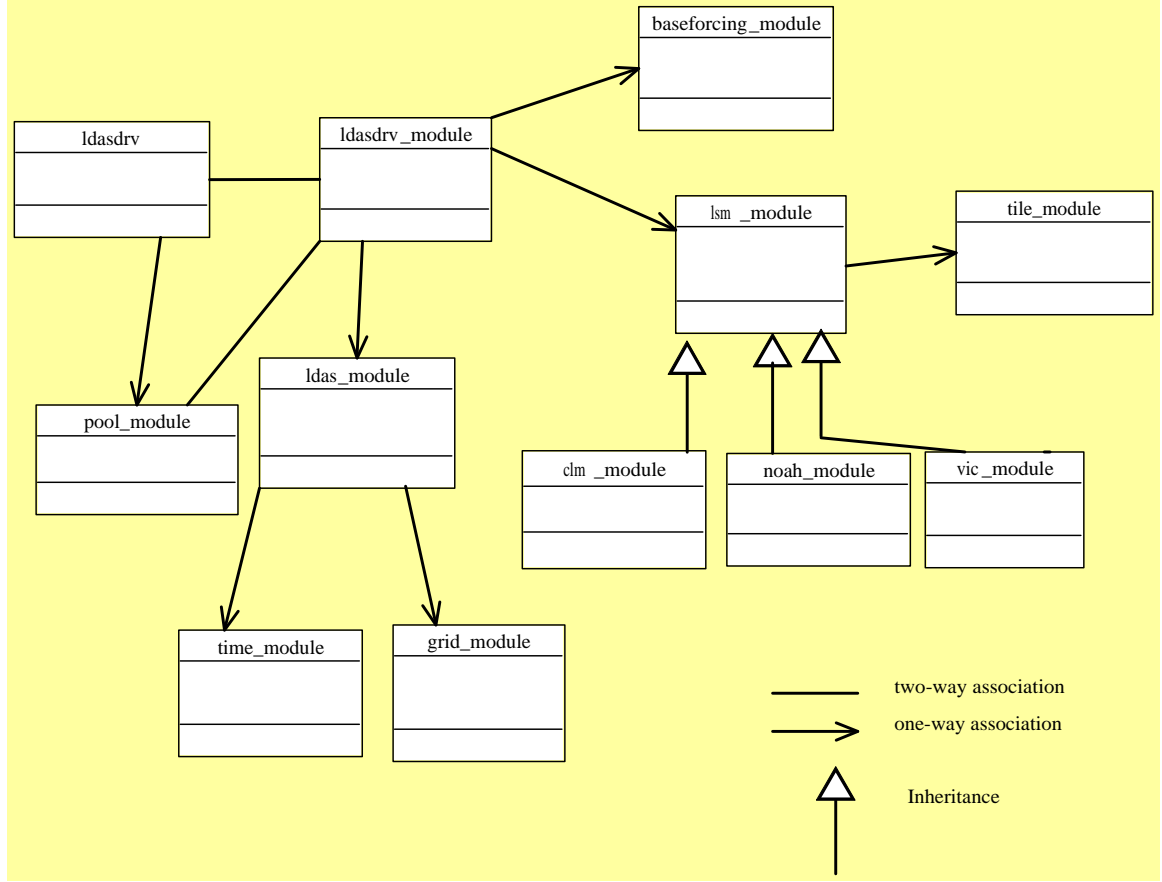


Figure 2: Structure of modules in LDAS driver

The design presented above is expected to evolve into a component-based framework. In component-based design, applications are constructed using software entities called components, which can be accessed only through predefined interfaces. The current interfaces in LIS will be implemented using function pointers to achieve a “component-like” outlook.

The design of LDAS driver presented above achieves encapsulation of data and control. The underlying representation does not need to be changed to incorporate a new forcing or a new LSM. The code also simulates polymorphism by allowing the initializations and executions to be determined at runtime. For example, `lsm_module` contains a global table of pointers for each LSM in the inheritance hierarchy. `lsm_module` acts as a polymorphic class, delegating the program flow based on the global pointer that is instantiated. This method also helps in facilitating defining operation of ensembles of LSMs in addition to individual LSMs. Together, these

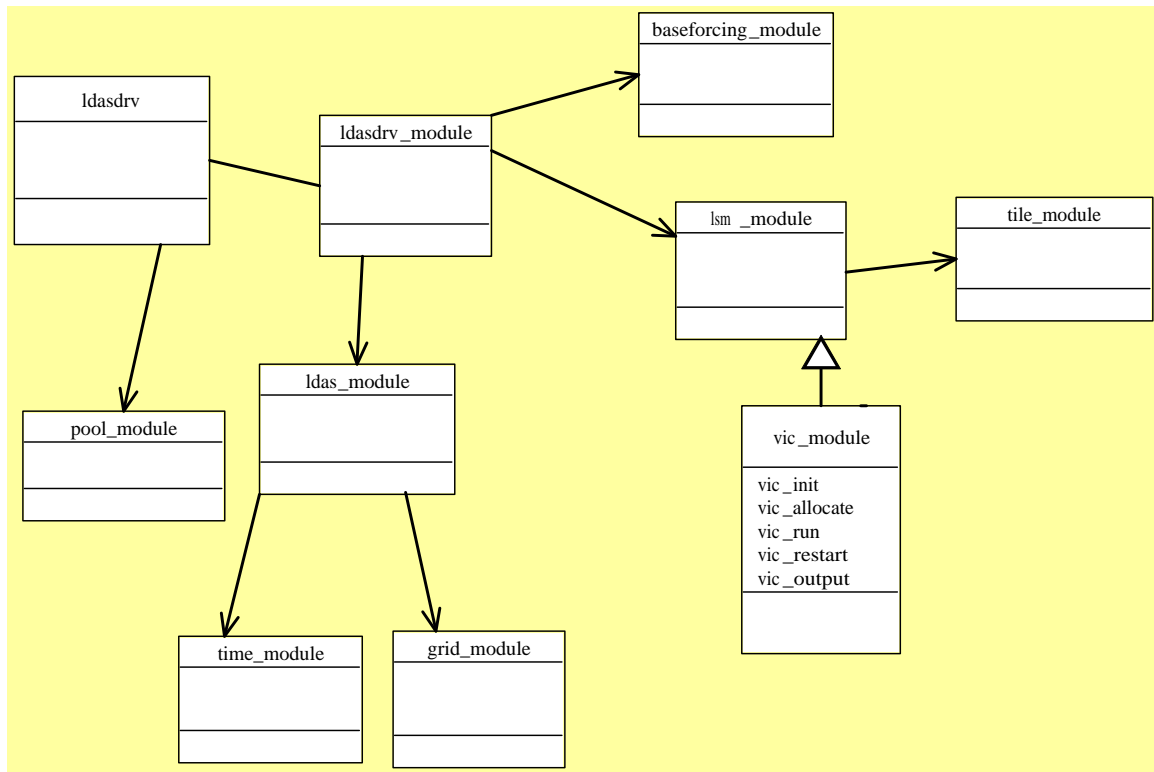


Figure 3: Example of introducing a new LSM(VIC) in LIS

concepts help to organize the code, making them more flexible, maintainable, and extensible.

4 External Code Interoperability in LIS

To demonstrate interoperability with other scientific modeling communities, LIS will comply with the ALMA data exchange convention and employ the utilities and extensible interfaces provided by ESMF.

4.1 ESMF

The purpose of ESMF is to develop a framework to enhance the ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, and data assimilation applications. ESMF is intended to provide a structured collection of building blocks that can be customized to develop model components. ESMF can be broadly viewed as consisting of an infrastructure of utilities and data

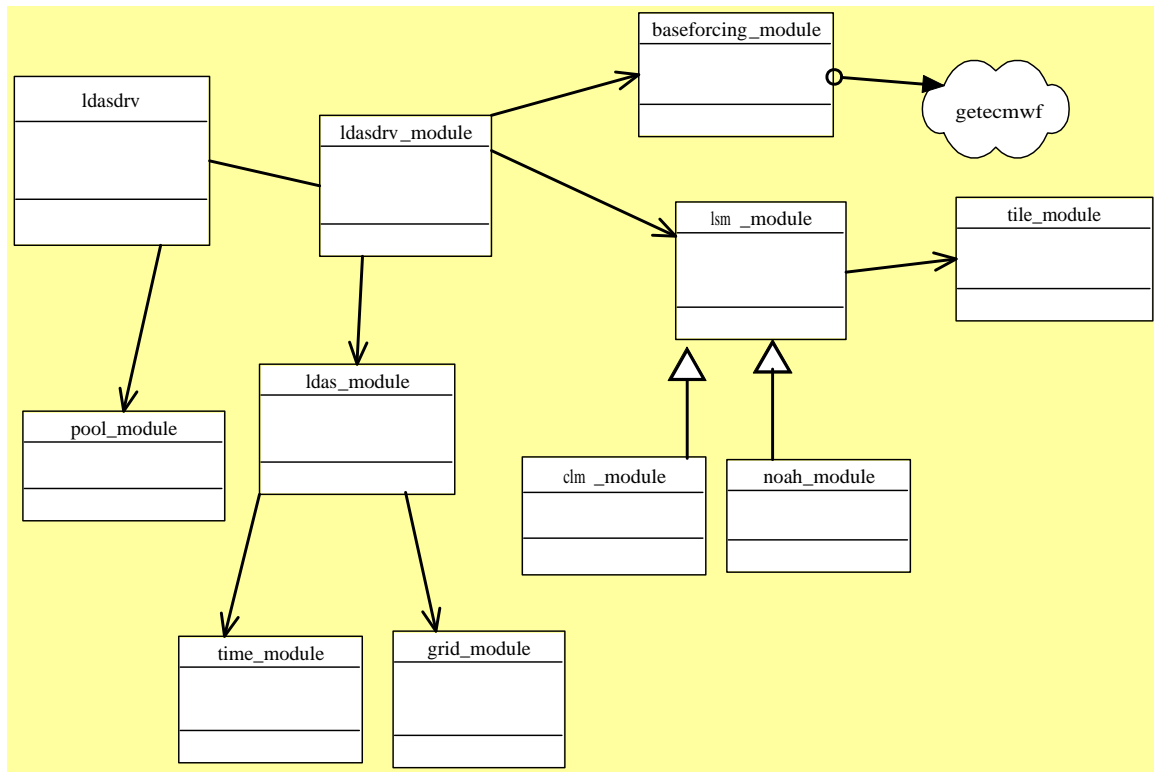


Figure 4: Example of introducing a new forcing(ECMWF) in LIS

structures for building model components and a superstructure for coupling and running them.

ESMF provides a utility layer that presents a uniform interface for common system functions. Some of the utilities include time manager, basic communications, error handler, diagnostics, machine model that provides abstractions of hardware and software, etc. LIS intends to use a number of these utilities as and when their implementations are complete.

ESMF also defines a number of guidelines for applications that are intended to be coupled. In order to achieve compliance, components must provide a minimal set of ESMF-defined interfaces. The implementation of these interfaces is expected to enable components to be initialized in parallel configurations, destroyed, and also provide query methods to extract pertinent information. LIS is expected to be a “gridded component” that provides the implementation of interfaces that use ESMF_State to exchange information with other models and systems. By implementing the ESMF interfaces at the LIS driver level, each LSM in LIS will not need to be customized to comply with the ESMF standards. LIS will also aim at implementing these interfaces

baseforcing_module.f	lsm_module
<pre> gdas :: private, integer, pointer geos :: private, integer, pointer eta :: private, integer, pointer ncep :: private, integer, pointer nasa :: private, integer, pointer ecmwf: private, integer, pointer </pre>	<pre> private, C :: clmdec, pointer clm (:) :: clmdec, pointer private, N :: noahdec, pointer noah(:) :: noahdec, pointer tile(:) :: tiledec, pointer </pre>
<pre> forcing_init(this, force) get_baseforcing(ldas,this,grid) </pre>	<pre> init(this, LSM) lsm _tile_allocate(this, nch) lsm _setup(ldas, this) lsm _init_output(ldas, this) lsm _green_alb (ldas, this) run_lsm _seq(t, ldas, this) run_lsm (pool, nch, ldas, this) lsm _readrestart(i, ldas,grid,this) write_output(ldas, grid, this) </pre>

Figure 5: Modules `baseforcing_module` and `lsm_module`

so that LIS can be coupled with other earth system models through ESMF as well as use the utilities provided for gridded components. Figure 6 shows a rough sketch of how applications will be initialized and coupled. By complying with the ESMF guidelines, LIS is expected to act as the land modeling component in ESMF.

In order to be ESMF compliant, at least three utilities provided by the ESMF Infrastructure layer need to be included. LIS currently uses the ESMF time management utility that provides useful functions for time and data calculations, and higher level functions that control model time stepping and alarms. The time management routines in LIS is implemented by using the ESMF time management functions. The `time_module` in Figure 2 contains functions that initializes, and delegates function calls to use the ESMF time management library functions.

The LIS milestone J (July 2004) will be the implementation of the ESMF compliant version of CLM in the LIS. The LIS plans to use several of the utilities before the 2004 timeframe such as the time management, the regridding and the communication services.

4.2 ALMA Interfaces in LIS

ALMA is a data exchange convention to facilitate the exchange of forcing data for LSM and the results produced by these schemes. The ALMA scheme enables inter-comparisons of land surface schemes and ensures that the implementation of proce-

```

type(ESMF_Comp) :: atm
type(ESMF_Comp) :: land           !replaced by an instance of LIS
type(ESMF_Comp) :: coupler
....
atm = ESMF_CompCreate('atm_comp', .. )
land = ESMF_CompCreate('lis', ....)
coupler = ESMF_CompCreate(...)
...
atm = ESMF_CompInit(atm)
land = ESMF_CompInit(land)
...

```

Figure 6: A skeletal example of applications in ESMF

dures to exchange data needs to be done only once. ALMA provides a list of variables needed to force LSMs and a summary of output variable definitions for LSM inter-comparisons.

By implementing the ALMA convention in the LDAS driver, LIS can exchange data with other land surface modeling systems that are also ALMA compliant. Further, it will enable the use of LIS for intercomparison of land surface models for high resolution global modeling.

In order for LIS to be ALMA compliant, a number of interfaces need to be defined as shown in Figure 7. The forcing data is fetched from various locations on the internet, and after preprocessing is fed to the LDAS driver, which in turns controls the execution of different LSMs. The input interface is expected to convert the forcing data into an ALMA compliant form. The ALMA wrappers for each LSM is expected to perform the translation of LDAS driver variables to the LSM variables. The output interface is intended to convert the outputs from various LSMs into an ALMA format. Various design issues for these interfaces are discussed below.

4.2.1 Input Interface

Global atmospheric model predictions provide baseline forcing for LDAS, but whenever possible, the modeled fields are replaced or corrected by observation-based fields. The global data are currently in various different data formats. The preprocessing routines for input data will convert the fetched data from internet into a self describing data format such as netcdf/grib. The Input interface will make use of the metadata information present in these data files along with the input forcing ALMA

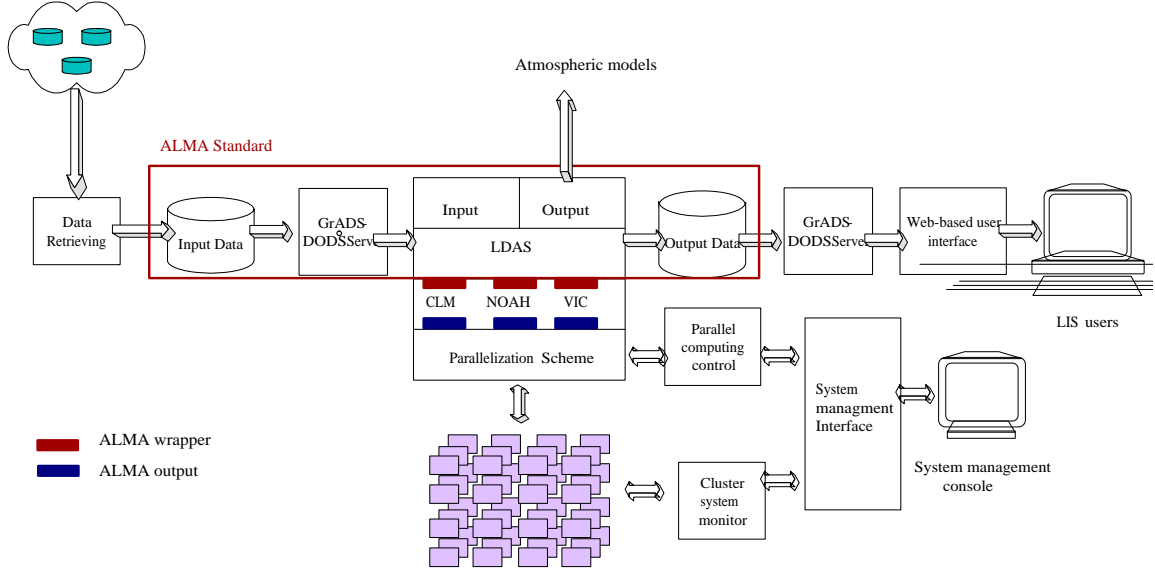


Figure 7: Proposed ALMA interfaces in LIS

definitions to generate an ALMA compliant format. Special attention need to be paid to the following issues.

- **Units** : All the required information to convert a forcing variable to the ALMA definition form need to be supplied. An exhaustive list of possibilities for the each forcing field need to be defined. Some special cases might also include dimensionless variables. For e.g, all the required information to convert a relative humidity value to a specific humidity value need to be specified.
- **Direction** : The sign conventions for each variable definition need to be converted to the ALMA format. Some forcing schemes might define a positive sign to be for an exchange from land to atmosphere, whereas another might consider positive sign to be for a downward direction from the sun to the earth. An exhaustive list of possible fields that need to convert a given directional definition to the ALMA format need to be specified.
- **Other issues**: The LDAS driver might require some variables (derived or otherwise) that does not fall within the current definition of input ALMA definition. The input interface will provide these variables. Compliance to the ALMA standard is considered to be providing all the variables that are specified in the definition.

Table 1: Mapping of Forcing variables to NOAH input variables

ALMA variable	Units	Sign (direction of positive values)	NOAH variable	Units	Sign	Required Conversion
<i>Wind_N</i>	$\frac{m}{s}$	Northward	<i>VWIND</i>	$\frac{m}{s}$	Northward	-
<i>Wind_E</i>	$\frac{m}{s}$	Eastward	<i>UWIND</i>	$\frac{m}{s}$	Eastward	-
<i>Rainf</i>	$\frac{kg}{m^2 s}$	Downward	<i>PRCP</i>	$\frac{mm}{s}$	Downward	
<i>Snowf</i>	$\frac{kg}{m^2 s}$	Downward				
<i>Tair</i>	<i>K</i>	-	<i>SFCTEMP</i>	<i>K</i>	-	-
<i>Qair</i>	$\frac{kg}{kg}$	-	<i>Q2</i>	-	-	-
<i>PSurf</i>	<i>Pa</i>	-	<i>SFCPRS</i>	<i>Pa</i>	-	
<i>SWdown</i>	$\frac{W}{m^2}$	Downward	<i>SOLDN</i>	$\frac{W}{m^2}$	Downward	-
<i>LWdown</i>	$\frac{W}{m^2}$	Downward	<i>LWDN</i>	$\frac{W}{m^2}$	Downward	-
<i>LSRainf</i>	$\frac{kg}{m^2 s}$	-				
<i>CRainf</i>	$\frac{kg}{m^2 s}$	-	<i>CPCP</i>	$\frac{mm}{s}$	-	
<i>CSnowf</i>	$\frac{kg}{m^2 s}$	-				
<i>LSSnowf</i>	$\frac{kg}{m^2 s}$	-				
<i>SVRainf</i>	$(\frac{kg}{m^2 s})^2$	-				
<i>SVSnowf</i>	$(\frac{kg}{m^2 s})^2$	-				
<i>Wind</i>	$\frac{m}{s}$	-	<i>SFCSPD</i>	$\frac{m}{s}$	-	-

4.2.2 ALMA Wrappers

Each LSM scheme included in LIS is expected to be capable of receiving variables in the ALMA form. The ALMA wrappers for each LSM will perform the required conversion from LDAS driver variables to LSM variables in accordance with the ALMA format. The design includes a list of conversions required. Table 1 includes a sample list a mapping of forcing data to NOAH variables. Similar tables will be required for other LSMs included in LIS.

4.2.3 Output Interface

Defining a generic output interface that converts output variables from different LSMs to an ALMA format is difficult, since explicit information is required to do the mapping from an LSM variable to a corresponding ALMA output variable. One of the intents of the ALMA standard is to put the onus of complying to the ALMA output standard on the LSMs so that intercomparisons between them can be done seamlessly. LIS will adopt this philosophy, assuming that the LSMs are ALMA compliant.

ALMA output standard lists a number of mandatory variables that are required

Table 2: Mapping of ALMA and VIC output variables : General Energy Balance

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>SWnet</i>	$\frac{W}{m^2}$	Downward	M	<i>net_short</i>	$\frac{W}{m^2}$	Downward	Y
<i>LWnet</i>	$\frac{W}{m^2}$	Downward	M				
<i>Qle</i>	$\frac{W}{m^2}$	Upward	M	<i>latent</i>	$\frac{W}{m^2}$	Upward	Y
<i>Qh</i>	$\frac{W}{m^2}$	Upward	M	<i>sensible</i>	$\frac{W}{m^2}$	Upward	Y
<i>Qg</i>	$\frac{W}{m^2}$	Downward	M	<i>grnd_flux</i>	$\frac{W}{m^2}$	Downward	Y
<i>Qf</i>	$\frac{W}{m^2}$	Solid to Liquid	R				U
<i>Qv</i>	$\frac{W}{m^2}$	Solid to Vapor	O				
<i>Qtau</i>	$\frac{W}{m^2}$	Downward	R				U
<i>Qa</i>	$\frac{W}{m^2}$	Downward	O	<i>advection</i>	$\frac{W}{m^2}$	Downward	Y
<i>DelSurfHeat</i>	$\frac{J}{m^2}$	Increase	R				U
<i>DelColdCont</i>	$\frac{J}{m^2}$	Increase	R	<i>deltaCC</i>	$\frac{J}{m^2}$	Increase	Y

to do water and energy balance. The output interface will use these variables to compute water and energy balance calculations for different LSMs. The output of recommended and optional variables will depend on the LSM employed.

A mapping between lists of ALMA output variables and VIC variables are presented in Tables 2 to 9. Similar to the input interface, other LSMs in LIS are also expected to provide mapping between their output variables and ALMA output variables.

The ALMA standard categorizes each ALMA variables into a priority category, which appears in Tables 2 to 9 under the heading Priority. The priority indicates whether the variable is mandatory(M), recommended (R), or optional(O), to comply with the standard.

The status category in Tables 2 to 9 indicates the current status of the ALMA variable in the land surface model. A yes(Y) indicates that the ALMA mandatory variable is currently output from the model. A no(N) indicates that the ALMA mandatory variable does not exist in the current output from the model and would require changes in the model code to calculate the variable. An unavailable(U) variable indicates that the variable is not part of the model output currently.

5 Final Remarks

The goal of LIS is to develop a leading edge land surface modeling and data assimilation system to support broad land surface research and application activities, to

Table 3: Mapping of ALMA and VIC output variables : General Water Balance

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>Snowf</i>	$\frac{kg}{m^2s}$	Downward	M				U
<i>Rainf</i>	$\frac{kg}{m^2s}$	Downward	M	<i>prec</i>	$\frac{mm}{hr}$	Downward	Y
<i>Evap</i>	$\frac{kg}{m^2s}$	Upward	M	<i>evap</i>	$\frac{mm}{hr}$	Upward	Y
<i>Qs</i>	$\frac{kg}{m^2s}$	Out of grid cell	M	<i>runoff</i>	$\frac{mm}{hr}$	Out of grid cell	Y
<i>Qrec</i>	$\frac{kg}{m^2s}$	Into grid cell	O				N
<i>Qsb</i>	$\frac{kg}{m^2s}$	Out of grid cell	M	<i>baseflow</i>	$\frac{mm}{hr}$	Out of grid cell	Y
<i>Qsm</i>	$\frac{kg}{m^2s}$	Solid to liquid	M				U
<i>Qfz</i>	$\frac{kg}{m^2s}$	Liquid to solid	M				U
<i>Qst</i>	$\frac{kg}{m^2s}$	-	R				U
<i>DelSoilMoist</i>	$\frac{kg}{m^2}$	Increase	M				
<i>DelSWE</i>	$\frac{kg}{m^2}$	Increase	M				
<i>DeslSurfStor</i>	$\frac{kg}{m^2}$	Increase	M				U
<i>DelIntercept</i>	$\frac{kg}{m^2}$	Increase	R				U

Table 4: Mapping of ALMA and VIC output variables : Surface State Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>SnowT</i>	<i>K</i>	-	M				U
<i>VegT</i>	<i>K</i>	-	M				U
<i>BareSoilT</i>	<i>K</i>	-	M				U
<i>AvgSurfT</i>	<i>K</i>	-	M	<i>surf_temp</i>	<i>C</i>	-	Y
<i>RadT</i>	<i>K</i>	-	M	<i>rad_temp</i>	<i>K</i>	-	Y
<i>Albedo</i>	-	-	M	<i>albedo</i>	-	-	Y
<i>SWE</i>	$\frac{kg}{m^2}$	-	M	<i>swq</i>	<i>mm</i>	-	Y
<i>SWEVeg</i>	$\frac{kg}{m^2}$	-	O	<i>snow_canopy</i>	<i>mm</i>	-	Y
<i>SurfStor</i>	$\frac{kg}{m^2}$	-	M				

Table 5: Mapping of ALMA and VIC output variables : SubSurface State Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>SoilMoist</i>	$\frac{kg}{m^2}$	-	M	<i>moist</i>	<i>mm</i>	-	Y
<i>SoilTemp</i>	<i>K</i>	-	R				U
<i>SMLiqFrac</i>	-	-	O				U
<i>SMFrozFrac</i>	-	-	O				U
<i>SoilWet</i>	-	-	M				U

Table 6: Mapping of ALMA and VIC output variables : Evaporation Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>PotEvap</i>	$\frac{kg}{m^2s}$	Upward	R				U
<i>ECanop</i>	$\frac{kg}{m^2s}$	Upward	R	<i>evap_canop</i>	$\frac{mm}{hr}$	Upward	Y
<i>TVeg</i>	$\frac{kg}{m^2s}$	Upward	M	<i>evap_veg</i>	$\frac{mm}{hr}$	Upward	Y
<i>ESoil</i>	$\frac{kg}{m^2s}$	Upward	M	<i>evap_bare</i>	$\frac{mm}{hr}$	Upward	Y
<i>EWater</i>	$\frac{kg}{m^2s}$	Upward	R				U
<i>RootMoist</i>	$\frac{kg}{m^2}$	-	M				N
<i>CanopInt</i>	$\frac{kg}{m^2}$	-	R	<i>Wdew</i>	$\frac{mm}{hr}$	-	Y
<i>EvapSnow</i>	$\frac{kg}{m^2}$	-	R	<i>sub_snow</i>	$\frac{mm}{hr}$	-	Y
<i>SubSnow</i>	$\frac{kg}{m^2s}$	-	R	<i>sub_canop</i>	$\frac{mm}{hr}$	-	Y
<i>SubSurf</i>	$\frac{kg}{m^2s}$	-	R				U
<i>ACond</i>	$\frac{m}{s}$	-	M				U

Table 7: Mapping of ALMA and VIC output variables : Other hydrologic Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>Dis</i>	$\frac{m}{s}$	-	O				U
<i>WaterTableD</i>	<i>m</i>	-	O				N

Table 8: Mapping of ALMA and VIC output variables : Cold Season Processes

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>SnowFrac</i>	-	-	O				U
<i>RainSnowFrac</i>	-	-	O				U
<i>SnowfSnowFrac</i>	-	-	O				U
<i>IceFrac</i>	-	-	O				U
<i>IceT</i>	m	-	O				U
<i>Fdepth</i>	m	-	O				U
<i>Tdepth</i>	m	-	O				U
<i>SAlbedo</i>	-	-	R				U
<i>SnowTProf</i>	K	-	R				U
<i>SnowDepth</i>	m	-	R	<i>snow_depth</i>	<i>cm</i>	-	Y
<i>SliqFrac</i>	-	-	R				U

Table 9: Mapping of ALMA and VIC output variables : Variables to be compared with remote sensed data

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
<i>LWup</i>	$\frac{W}{m^2}$	Upward	R				

help define earth system modeling interoperability standards, and to lead the effective application of high performance computing to high-resolution, real-time earth system studies. The framework oriented design of LIS presented in this document and the use and adoption of standards such as ESMF and ALMA helps in providing a platform for land surface modelers and researchers. The extensible interfaces in LIS helps to ease the cost of development of new applications. Utilities such as tools for high performance computing and data assimilation helps researchers in rapid prototyping and development. Further, participation in the standards laid out by ESMF also helps in coupling with other earth system models.

References

- [1] ALMA. <http://www.lmd.jussieu.fr/ALMA>.
- [2] CCA. <http://www.cca-forum.org/>.
- [3] CLM. <http://www.cgd.ucar.edu/tss/clm/>.
- [4] ESMF. <http://www.esmf.ucar.edu>.
- [5] GLDAS. <http://ldas.gsfc.nasa.gov>.
- [6] LIS. <http://lis.gsfc.nasa.gov>.
- [7] NOAH. <ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahlsn/>.
- [8] VIC. <http://hydrology.princeton.edu/research/lis/index.html>.